**8.1.2外围设备的信息交换方式**

1.程序查询方式:是一种最简单的输入输出方式，数据在CPU和外设之间的传送完全靠计算机程序控制。

优点：CPU的操作和外设的操作能够同步，硬件结构简单。

缺点：外设动作慢，程序进入查询循环浪费CPU时间，适用于小型处理机。

2.程序中断方式:外设主动通知CPU，准备送出输入数据或接收输出数据的一种的方法。当一个中断发生时，CPU暂停它的现行程序，转向中断服务处理程序，输入或输出一个数据。中断处理完毕后，CPU返回原来的任务。

优点：节省了CPU的时间。

缺点：硬件结构相对复杂，服务开销时间相对大。

3.直接内存访问（DMA）方式：

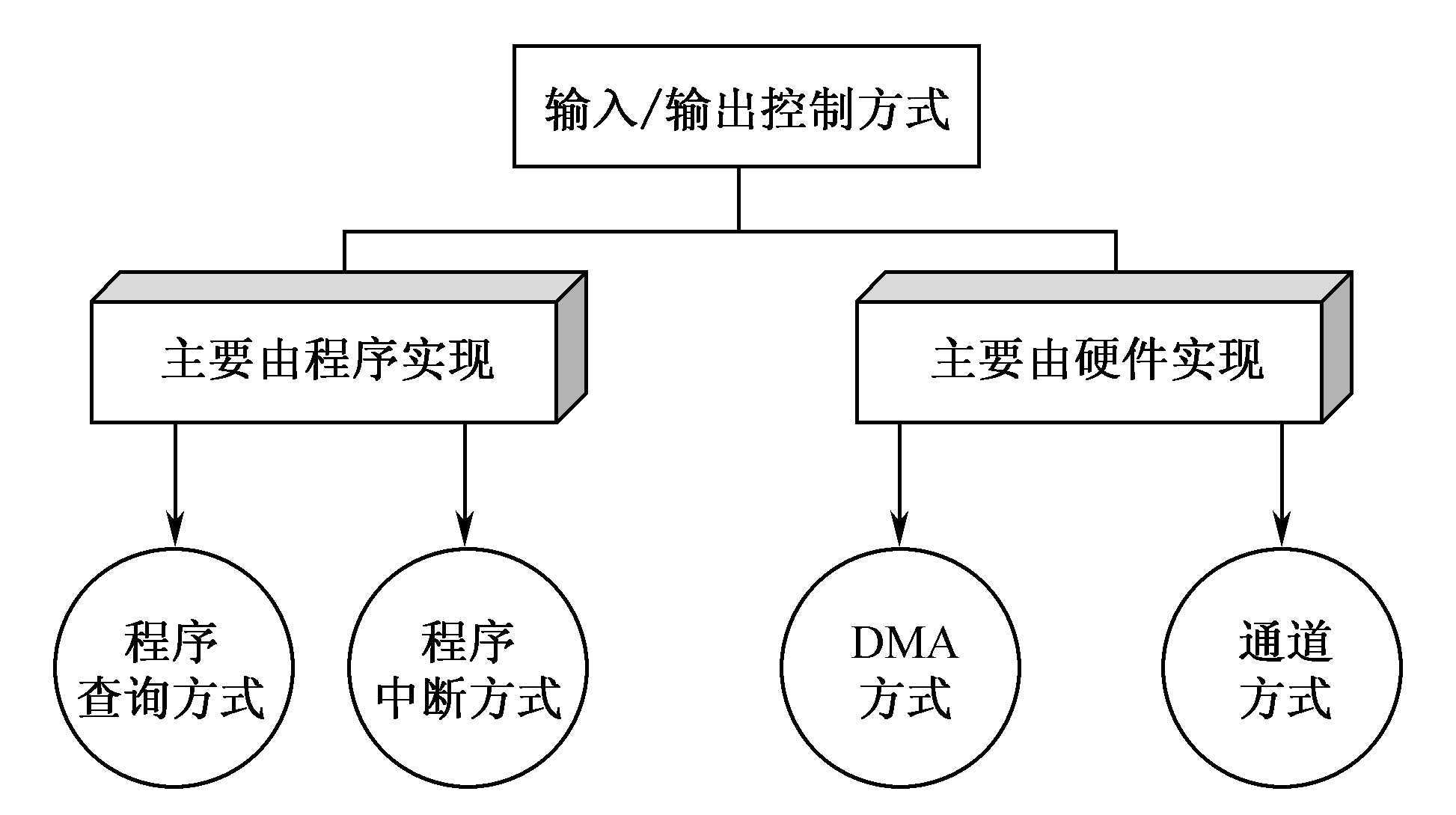
完全由硬件执行I/O交换的工作方式。DMA控制器从CPU完全接管对总线的控制，数据交换不经过CPU，直接在内存和外设之间进行，以高速传送数据。

特点：数据传送速度很高，传送速率仅受到内存访问时间的限制。但需要更多的硬件。DMA方式适用于内存和高速外设之间大批数据交换的场合。

4通道方式：通道是一个具有特殊功能的处理器，某些应用中称为输入输出处理器（IOP），它可以实现对外设的统一管理和外设与内存之间的数据传送。即CPU将部分权利下放给了通道。

特点：大大提高了CPU的工作效率，但这种提高CPU效率的方法是以花费更多硬件支持为代价的。适用于大型计算机。

外围设备的输入/输出控制方式：



**8.2程序查询方式**

1. 设备编址

统一编址：输入/输出设备中的控制寄存器、数据寄存器、状态寄存器等和内存单元一样看待，它们和内存单元联合在一起编排地址。

独立编址：内存地址和I/O设备地址是分开的，访问内存和访问I/O设备使用不同操作码的指令，即访问I/O设备有专门的I/O指令组。

2、输入输出指令

当程序实现输入/输出传送时，I/O指令一般具有以下功能：

（1）置“1”或置“0”I/O接口的某些控制触发器，用于控制设备进行某些动作，如启动、关闭设备的等。

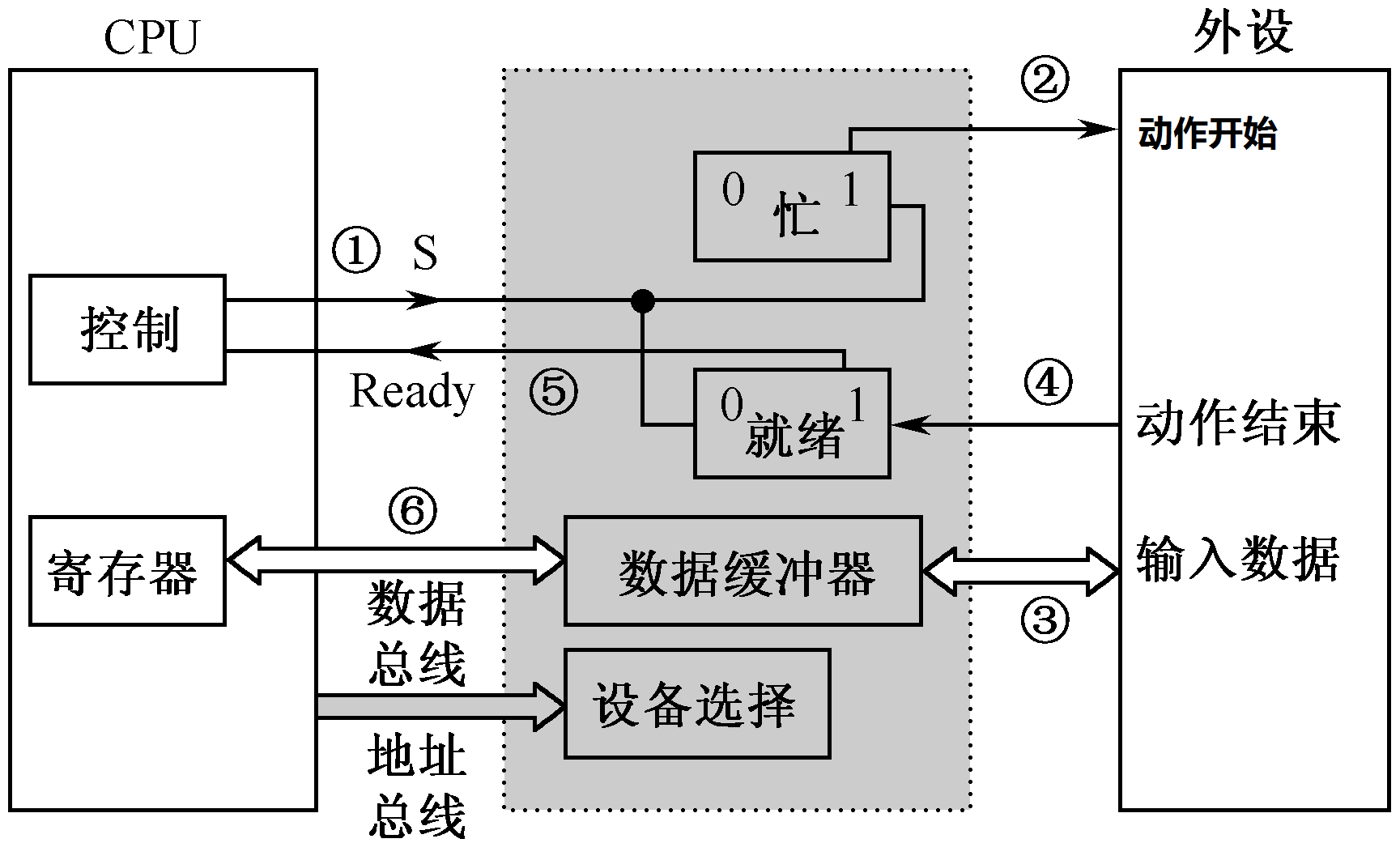
（2）测试设备的某些状态，如“忙”、“准备就绪”等，以便决定下一步的操作；

（3）传送数据，当输入数据时，将I/O接口中数据寄存器的内容送到CPU某一寄存器，当数据输出时，将CPU中某一寄存器的内容送到I/O接口的数据寄存器。

3、程序查询方式的接口

“接口”时总线与外设之间的一个逻辑部件，它作为一个转换器，保证外设用计算机系统特性所要求的形式发送或接收信息。

程序查询方式的接口如图所示：



程序查询方式的接口电路包括如下部分：

（1）设备选择电路：接到总线上的每个设备预先都给定了设备地址码。CPU执行I/O指令时需要把指令中的设备地址送到地址总线上，用以指示CPU要选择的设备。每个设备接口电路都包含一个设备选择电路，用它判别地址总线上呼叫的设备是不是本设备。设备选择电路实际上时设备地址的译码器。

（2）数据缓冲寄存器：当输入操作时，用数据缓冲寄存器来存放外设读出的数据，然后送往CPU;当输出操作时，用数据缓冲寄存器来存放CPU送来的数据，以便送给外设输出。

（3）设备状态：接口中的标志触发器，如“忙”、“准备就绪”、“错误”等，用来标志设备的工作状态，以便接口对外设动作进行监视。一旦CPU用程序询问外设时，将状态表示信息取至CPU进行分析。

4. 程序查询输入/输出方式

程序执行的动作：

(1) 先向I/O设备发出命令字，请求进行数据传送；

(2) 从I/O接口读入状态字；

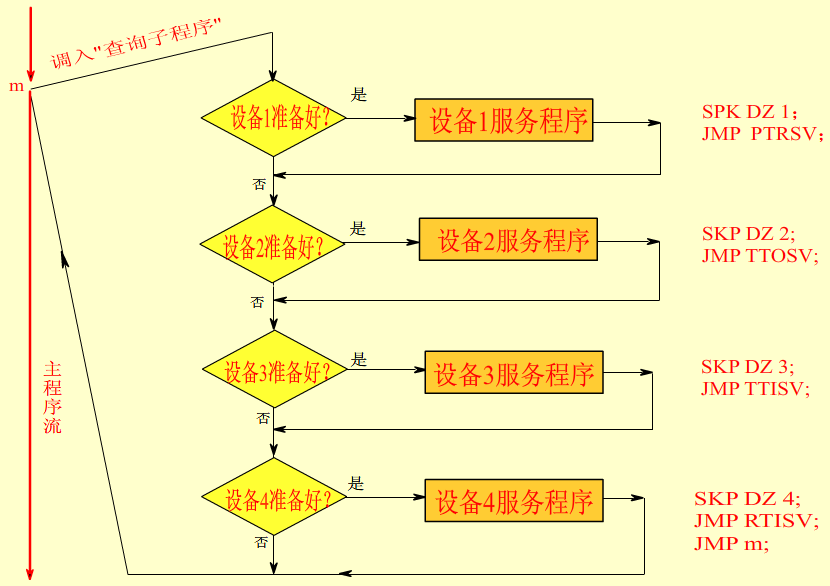
(3) 检查状态字中的标志，看看数据交换是否可以进行；

(4) 假如这个设备没有准备就绪，则第(2)、第(3)步重复进行，一直 到这个设备准备好交换数据，发出准备就绪信号“Ready”为止；

(5) CPU从I/O接口的数据缓冲寄存器输入数据，或者将数据从CPU输出至接口的数据缓冲寄存器。与此同时，CPU将接口中的状态标志复位;

(6) 数据传送。

缺点：CPU资源严重浪费。



**程序查询I/O设备流程图**

**程序中断方式**

**8.3.1中断的基本概念**

中断（Interrupt）：指CPU暂时中止现行程序，转去处理随机发生的紧急事件，处理完后自动返回原程序的功能和技术。

中断系统是计算机实现中断功能的软硬件总称。一般在CPU中设置中断机构，在外设接口中设置中断控制器，在软件上设置相应的中断服务程序。

中断处理过程注意几个问题：

（1）响应中断时机：外界中断请求时随机的，但CPU只有在当前指令执行完毕后，才转至公操作，即转入公操作（如中断处理、直接内存传送、取下条指令得）时才受理设备的中断请求。

（2）断点保护和现场保护问题：PC、相关寄存器内容和状态标志位都要保存到堆栈中，便于中断服务程序执行完毕后能够正确地返回断点处继续执行，并且恢复主程序里的相关参数。

（3）开中断和关中断问题：CPU响应中断后，执行中断服务程序之前，为了避免其他中断请求进来，此时要关闭中断；当CPU执行完中断服务程序后，要重新开放中断，以受理新的中断请求。

（4）中断是由软硬件结合起来实现的。

（5） 中断 内中断（异常）：机器内部原因导致出错引起的中断。

分为 外中断：外设请求服务的中断。

**8.3.6Pentium中断类型（中断响应的流程）**

**中断服务子程序进入进程**

中断服务子程序的入口地址信息存于中断向量号检索表内。实模式为中断向量表IVT，保护模式为中断描述符表IDT.

CPU识别中断类型取得中断向量号的途径：

（1）指令给出。如INT n中的n即为中断向量号。

（2）外部提供：可屏蔽中断是在CPU接收到INTR信号时产生一个中断识别周期，接收外部中断控制器由数据总线送来的中断向量号；非屏蔽中断是在接收到MNI信号时中断向量号固定为2。

（3）CPU识别错误、故障现象，根据异常和中断产生的条件自动指定向量号。

**3.中断处理过程**

1. 当中断处理的CPU控制权转移涉及特权级改变时，必须把当前的SS和ESP两个寄存器的内容压入系统堆栈予以保存。
2. 把CPU的标志寄存器进栈，保护各个标志位；
3. 清除IF和TF标志，屏蔽INTR中断和单步中断；
4. 当前的代码段寄存器CS和指令指针EIP也压入堆栈，即保护断点；
5. 如果中断发生伴随有错误码，则错误码也压入堆栈；
6. 从中断向量号获取的中断服务子程序的入口地址（段、偏移）

分别装入CS和EIP，开始执行中断服务子程序

（7）中断服务子程序最后的IRET指令使中断返回。保存在堆栈中的中断现场信息被恢复，并由中断点继续执行原程序。

**DMA方式**

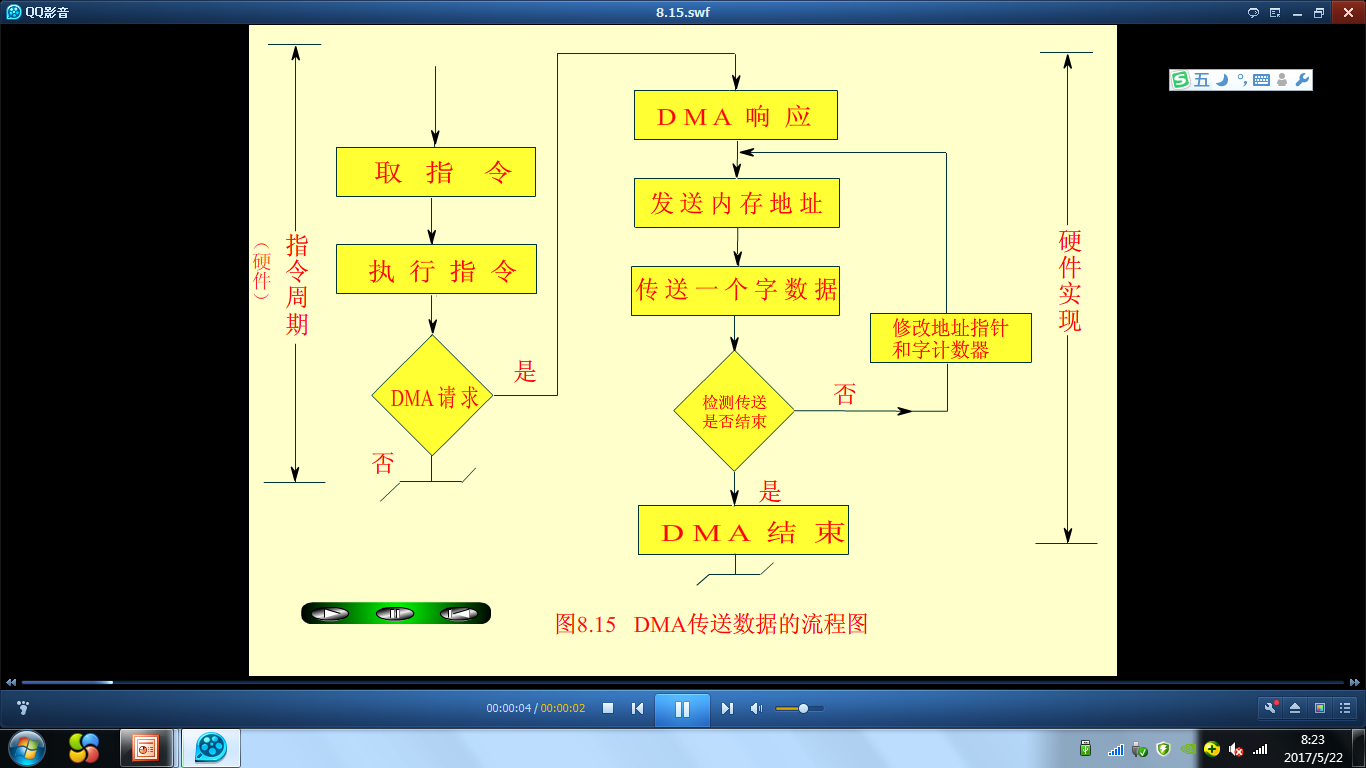
**DMA方式与中断的区别**

⑴中断方式是在数据缓冲寄存器满之后发出中断，要求CPU进行中断处理，而DMA方式则是在所要求传送的数据块全部传送结束时要求CPU 进行中断处理。这就大大减少了CPU进行中断处理的次数。  
⑵中断方式的数据传送是在中断处理时由CPU控制完成的，而DMA方式则是在DMA控制器的控制下，不经过CPU控制完成的。这就排除了CPU因并行设备过多而来不及处理以及因速度不匹配而造成数据丢失等现象。

**DMA的动作流程（没有在书本上找到这部分内容）**

DMA的数据块传送过程分为三个阶段：传送前预处理、正式传送、传送后处理。

1. 传送前预处理阶段由CPU执行几条输入输出指令，测试设备状态，向DMA控制器的设备地址寄存器中送入设备号并启动设备，向内存地址计数器中送入起始地址，向字计数器中送入交换的数据字个数。这些工作完成后，CPU继续执行原来的主程序。
2. DMA的数据块传送是以数据块为基本单位进行的，每次DMA控制器占用总线后，无论时数据输入操作，还是输出操作，都是通过循环实现的。当进入输入操作时，外设的数据（一次一个字或一个字节）传送内存；当进行输出操作时，内存的数据传向外设。
3. DMA的传送后处理：一旦DMA的中断请求得到响应，CPU停止主程序的执行，转去执行中断服务程序做一些DMA的结束处理工作。这些工作包括：校验送入内存的数据是否正确；决定是否继续用DMA方式传送下去；测试在传送过程中是否发生了错误等。



**8.5通道方式**

**8.5.1通道的功能**

执行通道指令，组织外围设备和内存进行数据传输，按I/O指令要求启动外围设备，向CPU报告中断等，具体有以下五项任务：

(1)接受CPU的I/O指令，按指令要求与指定的外围设备进行通信。

(2)从内存选取属于该通道程序的通道指令，经译码后向设备控制器和设备发送各种命令。

(3)组织外围设备和内存之间进行数据传送，并根据需要提供数据缓存的空间，以及提供数据存入内存的地址和传送的数据量。

(4)从外围设备得到设备的状态信息，形成并保存通道本身的状态信息，根据要求将这些状态信息送到内存的指定单元，供CPU使用。

(5)将外围设备的中断请求和通道本身的中断请求，按次序及时报告CPU。

**8.5.2通道的类型**

1.选择通道

选择通道每次只能从所连接的设备中选择一台I／O设备的通道程序，此刻该通道程序独占了整个通道。连接在选择通道上的若干设备，只能依次使用通道与主存传送数据

数据传送以成组（数据块）方式进行，每次传送一个数据块，因此，传送速率很高。选择通道多适合于快速设备（磁盘），这些设备相邻字之间的传送空闲时间极短。

1. 多路通道

<1>字节多路通道

字节多路通道是一种简单的共享通道，在时间分割的基础上，服务于多台低速和中速面向字符的外围设备。

<2>字节多路通道包括多个子通道，每个子通道服务于一个设备控制器，可以独立地执行通道指令。每个子通道都需要有字符缓冲寄存器、I／O请求标志／控制寄存器、主存地址寄存器和字节计数寄存器。而所有于通道的控制部分是公共的，由所有子通道所共享。通常，每个通道的有关指令和参量存放在主存固定单元中。当通道在逻辑上与某一设备连通时，将这些指令和参量取出来，送入公共控制部分的寄存器中使用。

<3>字节多路通道要求每种设备分时占用一个很短的时间片，不同的设备在各自分得的时间片内与通道建立传输连接，实现数据的传送。

2数组多路通道

数组多路通道把字节多路通道和选择通道的特点结合起来。它有多个子通道，既可以执行多路通道程序，象字节多路通道那样，所有子通道分时共享总通道；又可以用选择通道那样的方式传送数据。

**8.5.3**

**通道结构的发展**

通道结构的进一步发展，出现了两种计算机I/O系统结构，一种是通结构的I/O处理机（IOP），另一种是外围处理机（PPU）。

1、输入输出处理器（IOP）

IOP可以和CPU并行工作，提供高速的DMA处理能力，实现数据的高速传送。 但IOP不是一台独立的计算机，而是计算机系统中的一个部件。此外，有些IOP还提供数据的变换、搜索和字装配／分拆能力。

8位和16位微机中使用的Intel 8089 处理器就是这种通道型I／O处理器。

2、外围处理机

外围处理机结构更接近于一般处理机，或者就是选用已有的通用机。外围机基本上是独立于主处理机工作的，应用于大型高效率的计算机系统中。